E3MS: A traffic engineering prototype for autoprovisioning services in IP/DiffServ/MPLS networks

Xavier Hesselbach[1], Joan Antoni García-Espín[2], Miquel González[3,] Javier Gonzalo[4], Sergi Figuerola[2]

[1]Departamento de Ingeniería Telemática, Universidad Politécnica de Cataluña UPC. C/ Jordi Girona, 1 y 3, Mod. C3 – Campus Nord, 08034 Barcelona.

[2] Fundación i2CAT, C/ Jordi Girona, 1 y 3. Edificio Nexus. 08034 Barcelona.

[3] Tecsidel. Avda. Príncipe de Asturias, 43-45, 2º, Barcelona.

[4] Vodafone R&D. Software Lab. Huesca.

xavierh@entel.upc.edu, joan.antoni.garcia@i2cat.net, miquel.gonzalez@tecsidel.es, javier.gonzalo@vodafone.com, sergi.figuerola@i2cat.net.

***Abstract*- This paper presents the testbed definition, implementation and trials of a new strategy for traffic autoprovisioning for MPLS and IP/DiffServ. This is the proof of concept of a new scenario for traffic engineering, for self-configuring control and end-to-end quality of service management by means of a tool based on Web Services. The system is structured in 3 layers: A Graphical User Interface, a Network Elements layer (an interface to physical devices) and, in the middle, a Network Management System layer, where decisions about admission, load balancing, path selection, re-routing and bandwidth allocation per class are taken. The system includes Dynamic Resource Allocation (DRA) and Background Monitoring System (BMS) modules to globally manage network resources. The so-called Squatter and Legalization mechanisms are introduced as novelties added to traffic engineering. Those strategies permit the use of part of the available resources from other classes only while unused by the class owning them. The trials have validated the management system, using Cisco routers.**

***Keywords*- MPLS, autoprovisioning, traffic classification, dynamic resource allocation, rerouting.**

I. INTRODUCTION

Currently, MPLS (MultiProtocol Label Swithing) is considered one of the mechanisms that better performs the network service convergence of voice, video and data required nowadays, mainly due to its traffic engineering functionalities, which provides class of service tagging, traffic prioritization and resource optimization, by means of LSP (Label Switched Paths). However, in order to support the trend of IP towards the universal transport network - even in operator transport networks - new mechanisms to assure quality-of-service (QoS) need to be provided, since these networks have been so far designed for best effort traffic, which means that no guarantees are provided to the data flow. Up to now, in the best situation, quality is provided with limited QoS mechanisms.

DiffServ-over-MPLS architecture provides differentiated services with guaranteed QoS. For the end to end QoS guaranteed service provisioning, current IP networks need to be enhanced in availability and guaranteed QoS provisioning, although they offer flexibility and scalability. To guarantee the user-requested demands and to keep the network utilization as best as possible, the performance management of DiffServ-over-MPLS is essential. Therefore, there is a need to enhance MPLS network functionalities to fully support integrated mechanisms with DiffServ, providing automatic provisioning based on class of service mapping on queues, queues dimensioning and scheduling schemes. UMTS (Universal Mobile Telecommunications System) networks support four traffic classes: Conversational, Streaming, Interactive and Background, each of them requiring different QoS parameters. Conversational and streaming require end to end QoS to provide bandwidth, delay and jitter guarantees. Interactive and background classes are less strict for QoS parameters but even require service differentiation. The strategies presented on this paper will provide end to end QoS guarantees for UMTS services on MPLS networks [1], [2].

In order to achieve and provide the end-to-end QoS level required in the Next Generation Internet, this paper considers two essential schemes: (i) Differentiated Services (DiffServ) and (ii) MPLS-TE capabilities [3], [4], under a management tool architecture based on a Web Service approach for the automatic resource provisioning based on service differentiation, queue mappings, dimensioning and scheduling schemas. In this proposal, the QoS is provided by means of the strategy and actions taken by some of the E3MS subsystems: The CAC, the DRA and the BMS. The first one controls the admission of new calls only when resources are available (taking into account not only bandwidth but also delay, jitter and losses), the second one manages the resource allocation and set up parameters in the devices, and the third one reorganize the resources on an specific link or even along the whole network.

This paper is organized as follows: In the next section, we define the main concepts and goals. Next, the system architecture is presented. Section IV, V and VI introduces its main components: GUI, NMS and NEM. Section VII describes the strategy for admission control, and section VIII the so called Background Monitoring System. Next section summarizes a set of new strategies also included in the

testbed. Section X describes some selected trials done to validate the functionalities, and finally the paper concludes with the most important conclusions and some future works.

II. Definitions, goals and approach

The global scope of this work is to demonstrate the practical usefulness of the mechanisms of autoprovisioning in the future Internet, based on classification of IP traffic in classes of service from the DiffServ architecture and MPLS networks, creating a management and configuration system to make IP/MPLS-TE networks useful for operators in order to support the end to end QoS requirements that incoming UMTS services demand. The requirements put on the solution to be developed are:

- Configure the network from a central point in a friendly way, reducing the configuration time.
- The stability in the network configuration.
- Automate the reconfiguration of IP networks depending on the real traffic.
- Scalability in terms of number of routers that can be managed.
- On-line monitoring of network performance.
- Dynamic network resource allocation.
- Self-optimization of resources.
- Multi-manufacturer solution.
- Definition of different profiles for accessing the configuration and monitoring interface.

The management system provides QoS mechanisms at network elements to guarantee the SLA (Service Level Agreement). These mechanisms are:

- Traffic classification and metering: To identify and classify the traffic into different classes.
- Traffic marking: To mark the traffic, if necessary, and assign the matching DSCP value.
- Policing: To discard the traffic that doe not conform to the required policies.
- Load Balancing: To balance the load among different paths.
- Bandwidth reservation: To reserve the required bandwidth for a service class.
- Connection Admission Control: To admit or deny new traffic flows based in checking the available resources.

The system is able to make decisions itself about the optimum network configuration that must be used in each moment to obtain highest network performance.

III. System architecture and elements

The prototype Enigma3 Management System (E3MS) has been designed following a central, layered approach. Central architecture has been considered for concentrating the intelligence of the management plane in one unique entity. Nevertheless, this entity might be replicated for robustness and failure-proofing of the whole system. Layering has been taken into account to provide flexibility and modularity to E3MS.

The gluing between different layers has been implemented using Web Services and following a Service Oriented Architecture. This SOA/WS implementation enables E3MS as an open, advanced network resource provisioning service for the clients, that is, it is neither restricted to any provider/user platform nor technology dependent.

A. Layered architecture

E3MS is composed of four layers: User Interface (UI), Network Management (NM), Network Element (NE) and Physical Network (PN).

UI layer is composed by either Graphical User Interface (GUI) or Gateway entities. GUI is a piece of software adapting from standard HTTP browsing showed to human users to SOAP/WS used by NM layer. GUI also allows the user to comfortably configuring access-lists, policies, tunnels and routing within the PN, remotely.

When the user of E3MS is an external communication bus supporting either HTTP browsing or SOAP/WS communication model, a Gateway entity is used to translate the incoming requests to E3MS operation requests. From a practical point of view, the Gateway is not further than a translator, because no requests can be generated within E3MS to be sent towards the external bus.

Thus, E3MS is as an advanced network service, from the external bus point of view, since it provides advanced functionalities for self-management and planning. NM layer concentrates the major intelligence of E3MS. This piece of software is in charge of serving the provisioning requests coming from the user. Moreover, it handles advanced mechanisms for managing the QoS of the different LSPs established along the PN, as it will be described in the coming sections. NM layer is composed by the Network Management System entity, acting as the controller of the whole system, that is, the head of the hierarchical architecture.

NMS uses an abstract image of the network, which is periodically updated either polling the Network Element managers at the NE layer or performing actions as a response to alarms coming from them. NE layer is composed of multiple Network Element Manager entities, each one dealing with one or more routers at the PN layer. NEM entities perform configuration and polling tasks over routers. Moreover, NEM can handle different SNMP traps sent by the router when events happen at PN layer (e.g. link failure, lost signaling, interface overload, etc.).

In general, the NE layer accomplishes three functions: First one is concerned with collecting and writing values from/to the Physical Network routers. The router configuration is done via remote CLI (Command Line Interface) execution command, as most of the MIBs provided by routers assemblers are read-only, which disables configuration via SNMP.

On second place, the router eventually sends notifications (traps) to the NEM, informing it about exceptional events within the network nodes. NEM correlates alarms in order not to overload the NMS with irrelevant notifications via the Alarm WS located at the NMS. Finally, the NEM is also responsible for monitoring the network to guarantee QoS commitments, performing polling tasks via the Background Monitoring System (BMS).

The PN layer used in E3MS must support the following features:

- DiffServ capable [5].
- MPLS-TE enabled [6].

- SNMP to access to MIB objects.
- MPLS CoS Enhancements.
- Resource Reservation Protocol (RSVP).
- Terminal access agent to manage LLQ, WRED and CB-WFQ.
- Policing, shaping and metering.

B. *System interfaces*

As introduced in the previous section, E3MS has three interfaces: UI-NM, NM-NE and NE-TN and an external interface. Interface towards remote administrators (upper interface) is based on SOAP-in-HTTP and needs Graphical User Interface entity to be deployed at UI layer. Therefore, all the entities that compose E3MS are deployed in web services, providing simplicity and service flexibility. The system architecture is shown at the Fig.1.

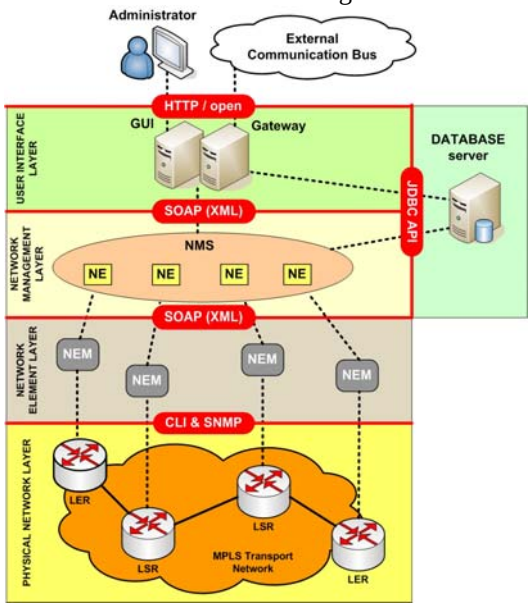

Fig. 1. MPLS-QoS management system architecture.

Interfaces between GUI-NMS and NMS-NEM are implemented using SOAP/XML, as commented before. Both are complaint with WSRF 1.2, as they are implemented with WSDL, deployed under Globus containers and using SOAP/XML. This strategy decouples in a high degree the different layers and eases their integration.

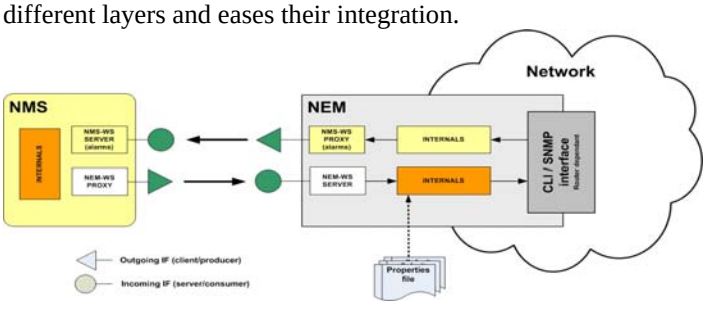

Fig. 2. NM-NE interface in detail. NEM and NMS provide services for network configuration and alarm, respectively.

NMS shows two services: The provisioning WS (NMS-WS) towards the GUI (UI-NM interface) and the alarm WS (Alarm-WS) towards the NEM (NM-NE interface). The NEM offers only one service: the router configuration service (NEM-WS). Fig.2 depicts the NM-NE interface.

Finally, the interface towards routers or network nodes considers two protocols: CLI-over-Telnet for executing remote command line operations and SNMP for either polling information or receiving alarm/events.

IV. The GUI implementation

The graphical user interface (GUI) in the Enigma E3MS system provides access to all the functionality of the NMS. It has been implemented as a web application, so network management can be done remotely and concurrently by multiple users.

A. *Technologies*

The GUI software consists of a standard web application, making use of standard open-source technologies such as:

- Apache Tomcat Server: Web application server (servlet and JSP container) [7].
- Java for implementation of classes supporting the business logic.
- JSP for presentation of dynamic web pages.
- Struts: Framework following the Model View Controler (MVC) architecture, for decoupling presentation from business logic [8].
- AJAX: Asynchronous Javascript And XML, for improving speed and usability in interactive web applications. It has been implemented using the DWR library [9].
- WebServices: for communication with the NMS, implemented with Globus Toolkit libraries.

B. *Functionality*

The network management functionality that can be accessed through the GUI includes the following:

- Node administration. Provides a way to create, modify and delete LER or LSR routers from the NMS inventory.
- Initial configuration of nodes. Allows the creation and modification of the initial configuration that is sent to the routers before any service provision is made. This initial configuration is divided in two steps:
  - Initial classes: Definition of service classes (from IPPrecedence0 to IPPrecedence5) and packet size for each one, which will be used by the initial policies.
  - Initial policies: Definition of policies applied in each interface sending MPLS traffic. For each service class, several parameters are defined such as bandwidth and queue size, and optionally RED and shaping parameters of traffic.
- Service provisioning. This is the main part, allowing the provision of services (MPLS LSP's) between any pair of LER's in the network. The available operations are creation, modification, rerouting and deletion of LSP's. In each operation the following parameters have to be introduced by the user:
  - General parameters such as LSP name, origin and destination, priorities.
  - Traffic sources, defining an access list for each class of service that will be injected into the LSP.

- Quality classes, defining the minimum quality requirements for each class of service, in terms of bandwidth, delay, jitter and packet loss.
- Routing, for choosing the LSP route calculation method, which can be explicit, OSPF or calculated by the CAC (Call Admission Control) algorithm.

- Network maps, showing a topological view of nodes and links in the network, marking the state of each element and allowing access to detail data about nodes and link occupation.
- Alarm console, for real time display of alarms generated by the NMS, related to network and service events.

Fig. 3. Main screen for service provisioning.

C. Communication with NMS

The GUI module is a client to the different WebServices offered from the NMS, and defined in a WSDL (WebServices Definition Language) contract. These services are accessible in a standard way from the GUI.

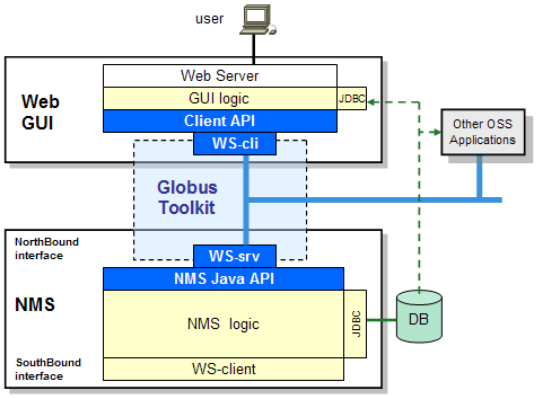

Fig. 4. Layered structure between GUI and NMS.

The client communication layer has been developed using the Globus Toolkit library, implementing the calls defined in the WSDL contract. It follows the IETF standard WS-RF (WebServices Resource Framework) for storing certain status variables between successive calls.

The implementation follows a layered architecture with clearly defined interfaces, which facilitates the independent development or even substitution of the composing parts shown in Fig.4.

V. The NMS

The Network Management Systems (NMS) module controls the main functionalities of the system, such as admission control, route selection, auto-rerouting and per-node resources allocation.

The NMS is in between the GUI module and the set of NEM modules. Therefore, it attends the demands coming from the GUI (administrator), and manages the resources. As a result, the NMS sends configuration requests to the NEMs, and so, to the physical devices.

Alarms and management messages are also processed by the NMS, in order to provide an integrated environment. Three elements define the main features:

- The NMS includes an image of the network topology, in order to reduce the amount of requests commands to the NEMs.
- Integrates a Call Admission Control strategy (submitted to the European Patents Office) and a Background Monitoring System.
- Controls the management network (LSPs and events and/or alarms).

VI. The NEM module

The Network Element Management is the module in charge of managing the routers of the network. The Network Management System invokes configuration or performance services to the NEM in order to perform desired operations on the devices. The NEM design has been done under two objectives: The first one is the communication with NMS and with routers in order to satisfy the needed requirements. The second consists on getting a hierarchy that can facilitate the NMS invocations treatment and the required planning in order to perform all processes that have to be done.

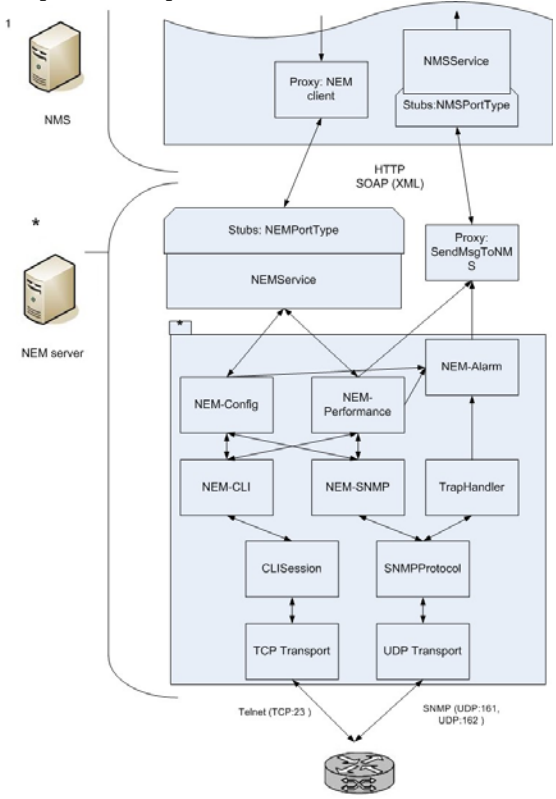

Fig. 5. NEM internal architecture overview.

In order to achieve these objectives, NEM has been designed following the class architecture shown in Fig. 5.
Messages arriving from NMS that invoke a Service of the NEM are received through a communication interface: NEMPortType, which uses stub translators implemented in Globus Toolkit 4 in order to de-serialize XML (SOAP) messages and translate them into Java operations. These operations are allocated on the main module: NEMService.

Once NMS has started, it calls the AddNEM service in order to create a new NEM instance for managing a specific router, sending into this request all parameters needed by the NEM in order to communicate later with the router. The first things that it will perform in this survey is the creation of all the objects needed in order to threat the NMS requests and also querying the router for his inventory (by SNMP) through the Configuration Module. Once the router has answered with its interfaces, then they are returned as a response for the AddNEM.

From there on, the NMS can invoke the different services defined on the NEMService and these class acts as a bridge calling the corresponding operation in the Configuration or Performance Modules and responding to the NMS once the operation has been finished.

Inside the NEM, 3 main modules have been defined:

1. The NEM-CONFIG is the responsible of orchestrate the router configuration concrete methods of the CLI module and its SNMP verifications in order to perform the complex requested operation. The main operations that this module can do are:
   - InitialClass: Configures or deletes Classes of Survey (CoS) on the router.
   - InitialPolicy: Configures one or several policy-maps in order to configure some of the previously configured CoS on corresponding router interfaces.
   - LSPRoute: Creates a new LSP on the source router of the desired Tunnel, obtaining the path by two possibilities: explicit route and implicit route. With explicit router the request has to contain all IP addresses of the nodes that conforms it. In the other hand, if it is implicit, the path is obtained by a dynamic routing protocol as OSPF.
   - Filter: With this operation, the type of traffic that has to be transported through the LSP can be defined. It configures into the router corresponding access-lists and it creates a route-map in order to redirect traffic classified by previous access-lists inside the LSP. Finally all this policy is applied to the corresponding LER interface. With this operation the traffic filtering can be created, modified or deleted. The mapping of the Survey Classes (CoS) defined into the interface with the corresponding access-list lines is done by IPPrecedence correspondence (because in the InitialPolicy a specific value has been assigned for each CoS configured in the interface).
2. The NEM-PERFORMANCE monitors different router parameters under NMS Monitoring requests and after obtaining the values from routers it is showing them and sending them, and also it can create alarms when it detects anomalies. Monitorings are based on measurements by SNMP requests but sometimes CLI commands are needed and they have to be parsed. After requested values have been obtained, they are stored or will be used in order to perform some calc. After Notification Time a set of calculated values in that Time Interval are sent to the NMS by a provideStatistics (service allocated into NMSPortType). The SendMsgToNMS module is needed in order to call that service. For each start Monitoring request, a scheduling process is created and it is executed every Monitoring Time, querying the needed values and storing corresponding results. There are the following types of Monitoring: LSP Traffic Rate, Interface Traffic Rate, Queue Delay, Jitter, Delay, Packet Losses, CPU usage and Bandwidth Utilization for Class inside LSP. The last type is the one used by the BMS Module of the NMS. It checks if BW configured for the different classes inside an LSP aren't infra-used or over-used. Thus, this module sends alarms to the Alarm module in case of one of these situations. The values obtained on each measurement execution are used for calculating a mean that will be used for the comparison with the configured BW in order to know if they are over or under the BW thresholds. If this situation occurs, then a message is queued into NEM-ALARM module that it will invoke the trapNotification Service of the NMS in order to communicate that situation.
3. The NEM-ALARM is the responsible of capturing and organizing the alarms that arrive from the router or also from NEM-PERFORMANCE and sent them to the NMS by the NMSPortType services invocation, using the SendMsgToNMS module.

In order to communicate with devices modules for configuration, query and monitor functionalities are needed, and also for retrieving alarms from them. In order to get it, the NEM supports different protocols: CLI and SNMP. CLI needs to use TCP as transport protocol and in order to configure devices by this way a TELNET session has to be opened. Current CLI interface communication is particular for Cisco's routers and CLI commands can be a little different with other vendors, but design and implementation have been done with modularity in order to be easily adaptable to other devices or vendors.

On the other hand, SNMP uses UDP as transport protocol. SNMP is used for monitoring routers and to do verifications for the configurations (sending messages through port 161) and also in order to receive alarms from the devices by TrapHandler module, that listens on port 162. To ensure that no process interferes other process operation (i.e. performing BMS operations when configuration operation is being done), a Mutual Exclusion System by occupation semaphores has had to be implemented. With them, two processes that send CLI commands to the same router can be executed at the same time and their respective responses from the router will not be mixed.

While the NEM application is being executed, many errors can be taken. So that, an error control system has been implemented in order to detect these situations and to solve them or to return a remote exception to the NMS with the error code and description.

In order to monitor the NEM execution while it is making configurations, an event logging system with several levels has been implemented using Apache Log4j.

There are a set of constants stored into a property file (XML format). The design of every part of the code can depend of the function of these constants when NEM application is executed.

VII. PER CLASS ADMISSION CONTROL

The E3MS controls the admission of new connections taking into account a specialized new CAC (Call Admission Control) strategy. This module, considers a set of classes. A new incoming call can be allocated in the own class of service, dropping a low priority connection, or even using resources of an upper class (when this resources are still unused). This last strategy is call squattering. The CAC algorithm takes decisions affecting to the routing. This can be on-line (when routing decisions are taken on the fly, according to the available information for the establishment of a path, without any additional optimization mechanism) or off-line (when a global research in the network is considered).

The general strategy proposed in this work and implemented is: For each LSP, each class is checked:

- First checks the available BW (bandwidth) for each class.
- If not available enough BW, try to establish as a squatter (in the upper class).
- If not available enough BW, try to provide BW by dropping lower priority LSPs.

The available routes considered (from a node A to a destination B) are:

1. The explicit route: the one provided – suggested - by the user. This is optional.
2. The route provided by the OSPF-TE protocol.
3. Rest of available routes: The set of routes can be build by means of an off-line research. Several protocols can be considered for this, such as Dijkstra. The result of this is a set of Potential Routes. Every route is considered potential route, because it still should be checked according to the QoS parameters demanded.

In order to allocate new connections, a Dynamic resource allocation (DRA) module can alter the per-class configuration of each node. This module is commented in the section IX.

VIII. THE BMS

The background monitoring system (BMS) is a specific module placed in the network management system (NMS) (see Fig.6). This module is responsible for listening the different notifications sent by the network element management (NEM) and, based on the information sent in every notification, apply some changes on the network when necessary. These notifications are state and error messages from the different routers of the MPLS network.

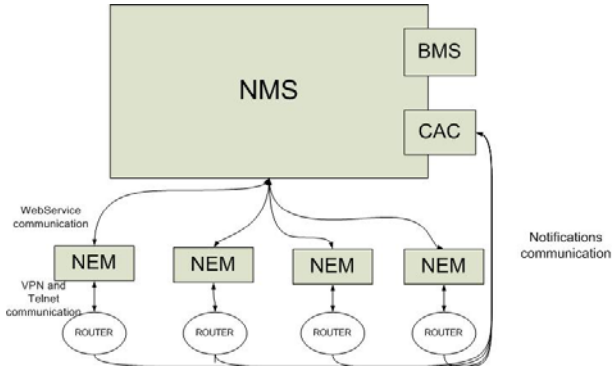

Fig. 6. Software modules involved in the BMS operation.

Actually, the two messages that trigger BMS working are the notifications about infra utilization and over utilization of bandwidth for a specific LSP. These two messages are sent when a LSP has reserved some bandwidth (say X Mbps) and the user specifies he wants to monitor this LSP. He specifies a maximum and minimum threshold percentage. If the router detects that the data traffic flow is under or over the specified percentage of the X Mbps reserved by the user, it sends a message of under or over utilization to the NEM and the NEM forwards this message also to the BMS.

First of all, the BMS checks if this message is the 5th message or more in order to permit the NEM to have a real value (the first 5 measurements normally are peaks values, not the medium value that it is the one that the BMS needs to perform real changes). The BMS then processes the information and tries to check if any change has to be applied on the network: the BMS set up this new bandwidth use measurement and convert this measurement into the next upper 8 multiple value (always upper because the goal is to permit this data flow). Once it has the next upper 8 multiple value (the routers in the testbed only supports reservations of 8 multiple values), it compares with the value reserved. If the value is different than the reserved by the user, the BMS tries to apply some changes on the network.

In the case of under utilization, a change is always applied because the action to be done is freeing resources and this is always possible to do. In the case of over utilization another module is needed to check if the changes can be applied, the CAC (this module checks if enough resources are in the network to apply the changes that the user/BMS wants to apply). BMS asks the CAC if the changes needed for this new measurement are possible and, if it is possible, BMS starts to apply the changes. The way to apply the changes is the same for the two cases, infra and over utilization: First of all the BMS updates the data base. Afterwards it sends a message to the NEM to stop the monitoring of this LSP in order to permit changing the original bandwidth reservation (the routers need to stop the monitoring to change a bandwidth reservation). Then, it sends another message to the NEM to modify the original reserved bandwidth with the new 8 multiple value. Finally, the BMS sends the last message to the NEM to perform the changes, the start monitoring for this LSP.

Consequently, there is a problem: If the BMS changes the state of the network, it is necessary to modify the network in mutual exclusion because if the BMS is changing the network status at the same time as a user request, both of them are changing a network that it is not consistent. For example, BMS changes the network status at the same time that the user asks the CAC to check if a LSP creation can be done. The CAC checks this creation in a network status that will be changed during the LSP creation, consequently this LSP is not going to be created in the network status the CAC thought that the network was. The solution is using a semaphore (see Fig.7) to access to the network state. When a user requests, for example, the creation of a new LSP, the NMS has to check if the BMS is changing at this time the network status (the routers configuration and also the data base). If the BMS is working on the network, the NMS has to wait to perform the LSP creation. Once the BMS has finished, the NMS can proceed by changing or consulting the network status depending on the user request.

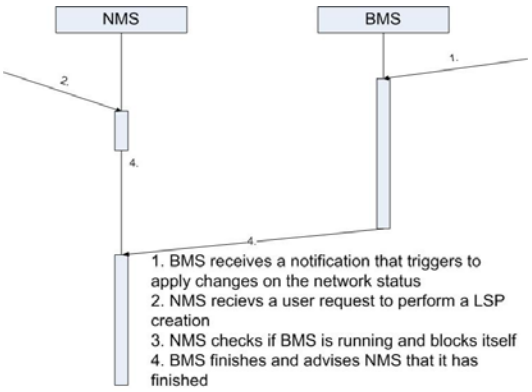

Fig. 7. NMS-BMS semaphore operation example.

While BMS is working, it blocks the rest of the NMS functions that can change/consult the network status.

IX. Other Functionalities

The E3MS implements other functionalities, in order to support per-node resources administration, taking into account classes of services: DRA, Marking, Load Balancing, Legalization and Off-line monitoring.

An enhancement of the nodes utilization can be applied by means of a specific control of per-class allocated bandwidth and queue sizes. This is made by means of the Dynamic Resource Allocation (DRA) module. This module can tune detailed performance parameters, such as per class queue length and schedulers in every node (router), according to the functionalities of every vendor. The NEM module translates the command according to acceptable messages in the device. Bandwidth and size of the queues created in output interfaces for each class should be changeable automatically by the CAC.

The Marking Module marks the routes that are showed in the compliant routes table from the most recommended to the least one. The criteria can be changed according to the administrator needs. In our testbed, we have considered a weighted formula, balancing the hop count, delay and remaining per-link bandwidth.

Besides, a Load Balancing strategy is implemented. This feature permits an enhancement in the network usage. Because the NMS have the knowledge about global network utilization, routing decisions are taken according to load balancing parameters, not only shortest path as in traditional networks such as internet using OSPF.

When some amount of BW is free in a queue (LSP deletion, rerouting, modification) it should be possible to legalize old squatter LSP. This mechanism is called legalization and is implemented in a Legalization module.

The Off-line Monitoring Module checks the status of the network in real time, monitoring statistics and looking for new and better network configurations. Some recommendations can be provided to the administrator by means of the GUI interface.

X. Samples of Trials and Validation

E3MS has been deeply validated over a testbed supporting multi-technology on the physical layer (from E1 radio links to WDM, also including Fast/Gigabit Ethernet). The testbed is composed of Cisco IP routing equipment from series 2800, 3600 and 3800. All routing devices are MPLS-TE capable.

For the tests shown in this paper, let be:

- LSP_name: name of an LSP.
- IPPi: IP traffic class. In the testing performed, an LSP carried traffic of several classes (IPP0-IPP2), being IPP2 the one with highest priority and IPP0 the less one.
- Route: sequence of routers crossed by an LSP. It is an array of output interfaces of each router involved in the path.
- QoS_bw_i: bandwidth demanded by each class carried by the LSP.

Moreover, for each class "IPPi" defined at an interface must be considered:

- Bw_original_i: portion of the bandwidth in a link which is reserved for class i. This allocation is not static, so it can change depending on the demanded bandwidth for any incoming LSP (DRA).
- Bw_used_i: aggregated bandwidth used for each class (by all LSPs). A distinction must done:
  - by a legal LSP: bandwidth used by the own class.
  - by a squatter LSP: bandwidth is used by another class j (and we say that class j is squattering class i
- Available_bw_i: bandwidth computed as Bw_original_i minus Bw_used_i.

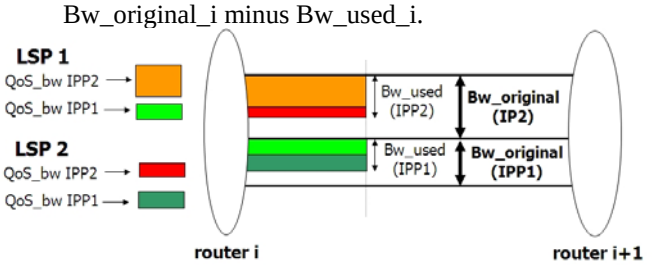

Fig. 8. A set of bandwidth specifications within E3MS.

A. Test 1. Squatter allocation and squatter LSP dropping

This test shows that, when there are not enough spare resources (in this case, bandwidth) in current class, E3MS can get bandwidth from another class. That is to say, if the desired bandwidth to be allocated is higher that the remainder from the original reserved for the class (remainder is equal to reserved minus currently allocated/in use), E3MS will logically establish the new LSP as a squatter LSP using part of the bandwidth reserved for another class. E3MS always tries to use the remainder resources in the legal class and add the necessary resources to establish the LSP by getting them from the spare ones in another class.

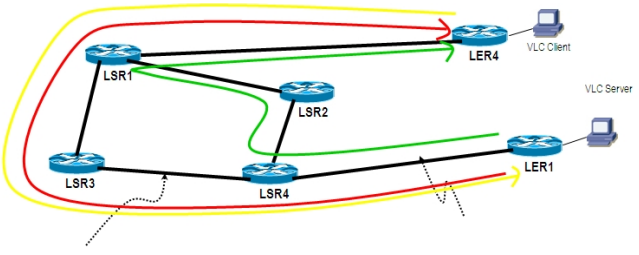

Fig. 9. Test 1 scenario.

Class	From LSR4 to LSR3			
	Original	Used		Available
		Legal	Squatt.	
Ipp2	800	0	0	800
Ipp1	800	480	0	320
Ipp0	200	16	0	194
Class	From LER1 to LSR4			
	Original	Used		Available
		Legal	Squatt.	
Ipp2	800	0	168	632
Ipp1	800	800	0	0
Ipp0	200	16	0	194

Table 1. Bandwidth allocation (kbps) in Fig.9 for Test 1.

The specifications and performance of this test are:

1. Tunnel41: 2 classes of traffic (IPP1 with 520 Kbps, IPP0 with 16Kbps) from LER1 to LER4. Route: LER1-LSR4-LSR2-LER1-LER4.
2. Tunnel42: 1 class of traffic (IPP1 with 448 Kbps) from LER1 to LER4. Route: LER1-LSR4-LSR3-LSR1-LER4. So IPP1 will be squattering IPP2 at output interface of LER1 and LSR1, as:
   a. it will get the 280 kbps available from IPP1 → so at output interface of LER1 and LSR1, Bw_used_1 (all legal) is 800 Kbps.
   b. it will get 168 Kbps from IPP2 → so at output interface of LER1 and LSR1, Bw_used_2 (all squatter) is 168 Kbps.
3. Tunnel14: 1 class of traffic (IPP1 with 96 Kbps) from LER4 to LER1. Route LER4-LSR1-LSR3-LSR4-LER1.
4. Video is transmitted correctly through tunnel42 despite of tunnel42 has not enough resources for IPP1 at output interface of LER1.

B. *Test 2. Squatter allocation and squatter LSP dropping*

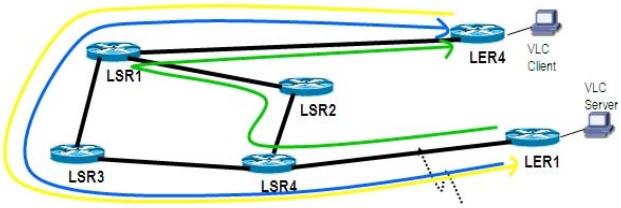

Fig. 10.Test 2 scenario.

Class	From LER1 to LSR4			
	Original	Used		Available
		Legal	Squatt.	
Ipp2	800	696	0	104
Ipp1	800	520	0	280
Ipp0	200	16	0	194

Table 2. Bandwidth allocation (kbps) in Fig.10 for Test 2.

The specifications and performance of this test are:

Tunnel43: 1 class of traffic (IPP2 with 696 Kbps) from LER1 to LER4. Route: LER1-LSR4-LSR3-LSR1-LER4, so squatter LSP2 will be dropped (as it is squattering IPP2 at output interface of LER1 and LSR1). So at output interface of LER1 and LSR1:

a. Bw_used_2 is 696 Kbps (all legal).
b. Bw_used_1 is 520 Kbps (all legal).
c. Bw_used_0 is 16 Kbps (all legal).

A lot of other validation tests have been successfully done, but they are not included in this paper.

XI. Conclusions

In this paper, the E3MS system has been introduced, described and analyzed. This system enables the operators to provide an end to end QoS auto-provisioning across an MPLS-Diffserv transport network, as a solution to support the new services requirements demanded by the Mobile Operators. It integrates a connection admission control and routing algorithms within the Network Management System in an innovative way to optimize the network resource allocation depending on the real traffic distribution and network performance. It also provides an intuitive interface to manage and configure the network, catching alarms and monitoring the traffic and the network performance.

The system design is modular, so new modules or components can be easily added to the system to provide other features in the future.

The results coming from project open a lot of further works to be done: Mainly, an analytical study to show the performance, scalability and the reliability of the system working on real scenarios. Also, the numerous variable values that have been introduced in the control algorithms must be tuned in order to work optimally in different real scenarios. We also consider the extension to routers from other vendors, such as Juniper Networks.

Acknowledgements

This work has been partially supported by the national spanish Project CICYT TSI2007-66637-C02 and the Enigma3 project from i2CAT foundation and Vodafone.

References

[1] 3GPP.TS23.107V6 UMTS; QoS Concepts and Architecture, TS23.107V6, March 2004.
[2] 3GPP.TS29.207V6 UMTS; QoS Concepts and Architecture, TS29.207V6, September 2004.
[3] S.Blake et.al. "An Architecture for Differentiated Services", RFC 2475, Dec. 1998.
[4] E.Rosen, A.Viswanathan and R.Callon, “Multiprotocol Label Switching Architecture”, RFC 3031, January 2001
[5] F. Le Faucheur, “Requirements for Support of Differentiated Services-Aware MPLS Traffic Engineering”, RFC 3564, June 2003.
[6] I.Minei, “MPLS DiffServ-Aware Traffic Engineering”, Juniper, Published 2004, Posted March 30, 2005.
[7] http://tomcat.apache.org/
[8] http://struts.apache.org/
[9] http://directwebremoting.org/dwr